[DM-04-077] Spatial Joins

Abstract

The measuring (or query) of the relationship between spatial features is of particular utility within a GIS. A spatial join combines represented geographic objects and their associated attributes based on a spatial relationship test (or predicate). The method of spatial join operation utilized depends on the relationship between the features represented and how those features are represented in the GIS. Regardless of the software implementation, the spatial join operation results are predicated on a test condition such as adjacency, proximity, or topology comparison among represented geographic data. This topic discusses how spatial join operations can be utilized for different geographic problems.

Keywords: joins, spatial queries, spatial relationships, tables

Author & citation

Morgan, J. D. (2023). Spatial Joins. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2023 Edition), John P. Wilson (Ed.).
DOI:10.22224/gistbok/2023.1.1

Explanation

1. Definitions

Minimum bounding rectangle: the smallest rectangle that encloses the object.

Spatial join: combines represented geographic objects and their associated attributes based on a spatial relationship test (or predicate).

Spatial predicate: uses a spatial relationship in a query to find objects that satisfy logical output conditions.

Table or non-spatial join: a database operation that combines two or more datasets based on common fields or columns

2. Introduction

The outcome of a geographic information systems (GIS) workflow is a set of spatial features or raster often displayed as a cartographic visual or map. However, the map as an end product only partly considers what is capable with GIS. The measuring (or query) of the

relationship between spatial features is of particular utility. A spatial join combines represented geographic objects and their associated attributes based on a spatial relationship test (or predicate). The method of spatial join operation utilized depends on the relationship between the features represented and how those features are represented in the GIS.

Geospatial software provides spatial join implementations, such as Esri's ArcGIS, PostgreSQL extension of PostGIS, R, and several libraries in Python (e.g., GeoPandas). Regardless of the software implementation, the spatial join operation results are predicated on a test condition such as adjacency, proximity, or topology comparison among represented geographic data. This chapter discusses how spatial join operations can be utilized for different geographic problems.

3. Non-spatial or Table Joins

Conventional database management systems use conceptual data model specifications to formalize design, facilitating design rigor, management, and communication. These data models may be implemented logically or physically, as described in Nyerges (2017a) and Nyerges (2017b), respectively, and include entity-relationship diagrams (ERDs) and unified modeling language object class diagrams. Since the advent of relational database systems (Codd, 1980), the operation of combining two different datasets based on a shared attribute is called a join (Worboys & Duckham, 2004).

To illustrate the relationship between data entities and the ability to join based on a common field (which can be multiple fields), consider the example of property parcels identified by parcel identifier numbers (pin) and the associated sales resulting in the exchange of ownership. Fig. 1 shows an ERD of the relationship between two entities (parcels and sales) related through the action of a selling or property exchange.

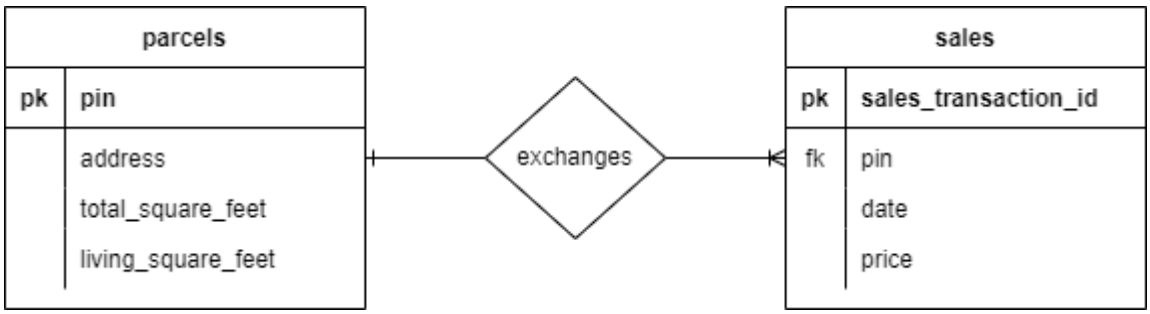

Figure 1. Example entity-relationship diagram (ERD) for table joins. Source: author.

In this case, the relationship between the parcel and the sales entity is specified as a one-to-many relationship because a single property can be exchanged through sales one or more times. See Nyerges (2017c) for a further discussion of the concept of cardinality entity classes. Given that Fig. 1 is a conceptual data model, a join operation is implemented in a database instance where a binary operator (string or mathematical matching) takes

two relations as an input and returns a single relation. Worboys & Duckham (2004) utilize relational algebra expressions to define the conceptual implementation of a join with the following syntax:

att1 = att2(rel1,rels)

where relations rel1 and rel2 are joined on the attribute combinations att1 of rel1 and att2 of rel2 and are the symbol for a natural join. The software implementation of a join can be carried out in desktop GIS or Structured Query Language or SQL. See Hachadoorian (2019) for more on SQL. For example, a SQL join carried out on the relations depicted in Fig. 1 that seeks to obtain the total square footage and sales price organized pin might look like this:

```
SELECT parcels.pin, parcels.total_square_feet, sales.price

FROM parcels

JOIN sales

ON parcels.pin=sales.pin

ORDER BY pin;
```

Notice how the foreign key (fk) in the sales dataset is joined to parcels using the primary key (pk) of the pin in the parcel's dataset. The above example is not exhaustive in demonstrating the types of table joins possible. It demonstrates how the operation utilized shared values in two different entities to fulfill the conditions of a predicate. Continuing the parcels/sales example, the relational algebra for this join would look like this:

pin=pin (parcels,sales)

Also, the attributes between the two entities classes combine in a single query based on the join operation. The term join can include the ability to relate two different datasets based on spatial relationships, which is the subject of the remainder of this chapter.

4. Spatial Joins

The conceptual basis for spatial join is to join attribute data based on a spatial relationship. Perform a spatial join by applying a predicate constraint to two spatial datasets. Egenhofer et al. (1999) and Mamoulis (2011) distinguish spatial predicates from join operations used in conventional databases. Spatial predicates utilize a spatial relationship in a query to find objects satisfying the predicate condition. Spatial join methods vary depending on the data model utilized in GIS. For instance, vector data includes points, lines, and polygons (Diamond, 2019). In the vector model, data points are represented as a single coordinate pair and used to represent spatial features with neither length nor area (0 mathematical dimensions or 0D). A line consists of an ordered list of point coordinate pairs and does have the measure of length (1D). A polygon usually represents a geographic feature with an enclosed area, where the first and last coordinate point is the same and can be used to measure the area and perimeter of a geographic feature (2D). Raster data is stored as an

array of georeferenced cells. The commonly used raster formats are just a single 2D array of pixels (Williams, 2019). Spatial join operations should be carried out on data utilizing the same coordinate projection for the most accurate results.

4.1 Based on Distance

Spatial joins can be performed based on proximity to and between features by setting a distance threshold such as a search radius parameter. For instance, spatial predicates for distance can utilize Boolean constraints such as a distance threshold (e.g., within 250 feet). Alternatively, the application of a distance-based spatial predicate can join geographic features near or within a certain distance. Here it is helpful to consider that the point is the simplest form of representing locations and that more complex forms of recording locations (e.g., polygons) can be approximated by points (Beeri, 2004). Also, lines are made up of a series of vertices or points.

Consider the case of comparing vehicle crash locations to the nearest roadway intersections. Adopting Calkins (1996) expanded ERD symbology which accounts for spatial relationships, allowing us to represent the spatial association between entities. Fig. 2 illustrates the spatial relationship between the two entities, vehicle_crash and street_intersection.

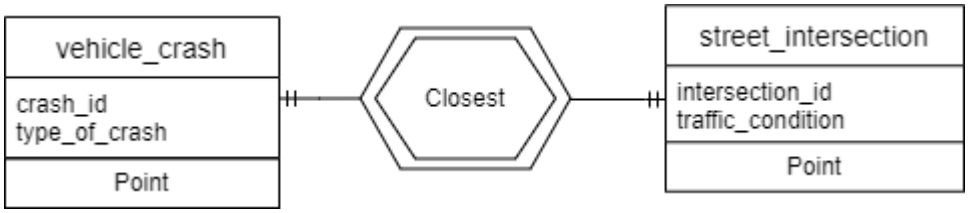

Figure 2. Example entity-relationship diagram (ERD) for a nearest spatial join. Source: author.

Notice that we specify a one-to-one cardinality here because the spatial predicate between street_intersection and intersection is specified as the closest or nearest. The spatial predicate can be carried out using Euclidean or straight-line distance, though network distance may be preferred in some GIS implementations. We can also conceptualize the spatial predicate in this example as a relational algebra expression:

closest(street_intersection, vehicle_crash)

Since the roads are typically stored in line format, a conversion to a point classified as the intersection will be part of the workflow to get them into a format for calculating distance. Once the intersections and the crash data are stored in point format, and both in the same projected coordinate system, a spatial join to query which intersection is closest to a given crash can be performed. Fig. 3 illustrates a spatial join based on the nearest or closest between target features (street_intersection) and source features (vehicle_crash).

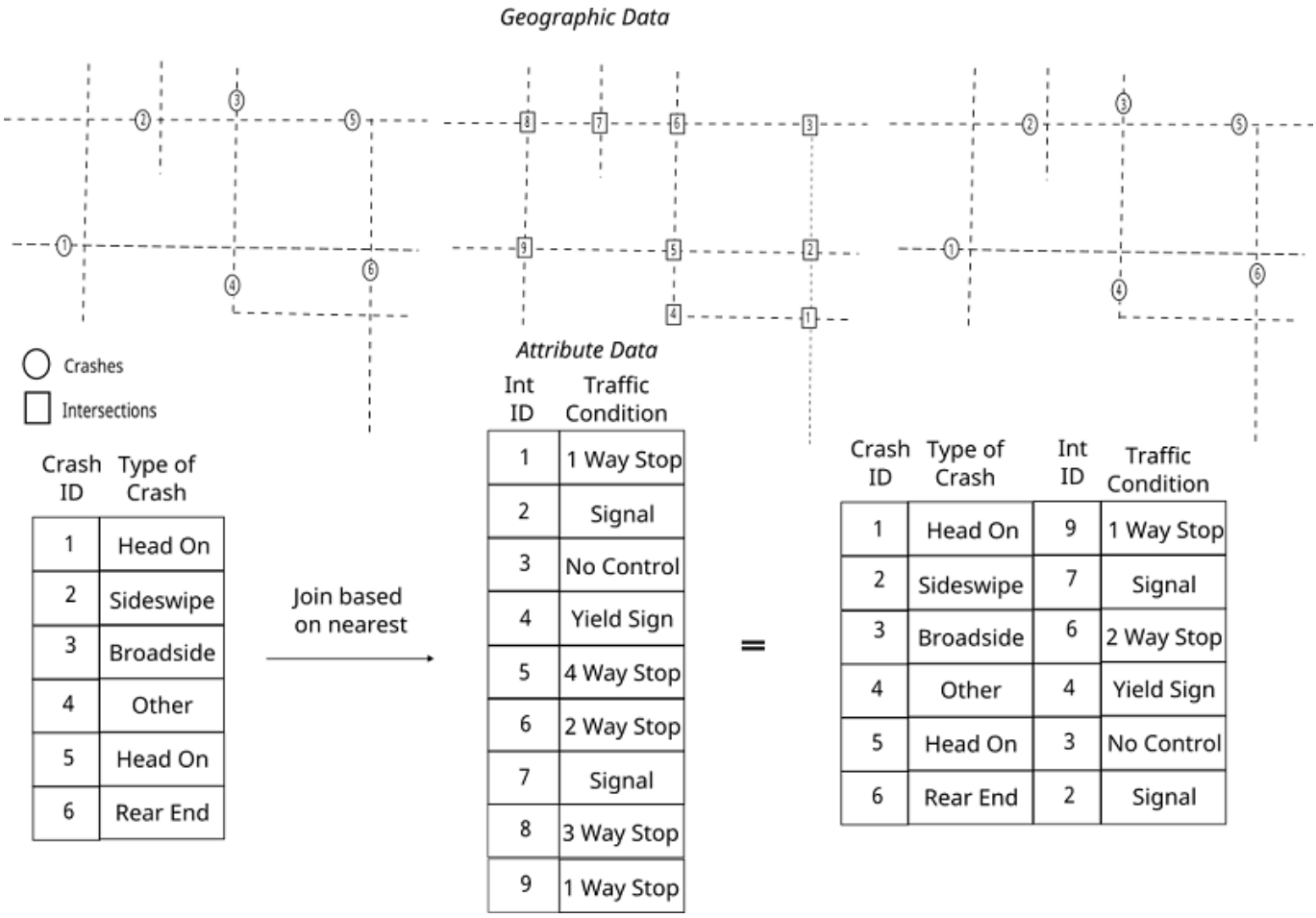

Figure 3. Example concept for a nearest spatial join. Source: author.

4.2 Based on Topology

Chan & Ng (1997) described a predicate as a logical function that returns true or false based on an output condition such as equal, inside, overlap, and adjacent. These topological operations can be the union or subtraction of features based on the overlay operation and result in a new dataset (Egenhofer et al., 1999). Cai (2022) notes that some of the most used vector overlay operations include intersection, union, erase and clip operations. Overlay operations perform set operations and are often helpful in steps leading up to a computed final GIS or feature data. When spatial data is at the extent that it is beyond the needs for a specific type of analysis, overlay operations can pare down data to the extent applicable to the analysis.

4.2.1 Polygon to Polygon

A typical GIS workflow calls for joining or spatially comparing relationships between geographic features represented as polygons. In this case, the join operation is carried out as a spatial predicate, such as a test across the features that checks for points that fall within the bounds of a given polygon or set of polygons. Topological properties such as overlaps, crosses, and intersects are relevant to spatial joins of this type.

Because polygons are 2D and made up of a series of lines and points, spatial joins of this

type are more computationally intensive. Consider the geographic case of determining which residential parcels will be assigned to which school zone. Here is a spatial join where the predicate specifies the largest amount of geographically represented overlap; the result might look like those conceptually illustrated in Fig. 4.

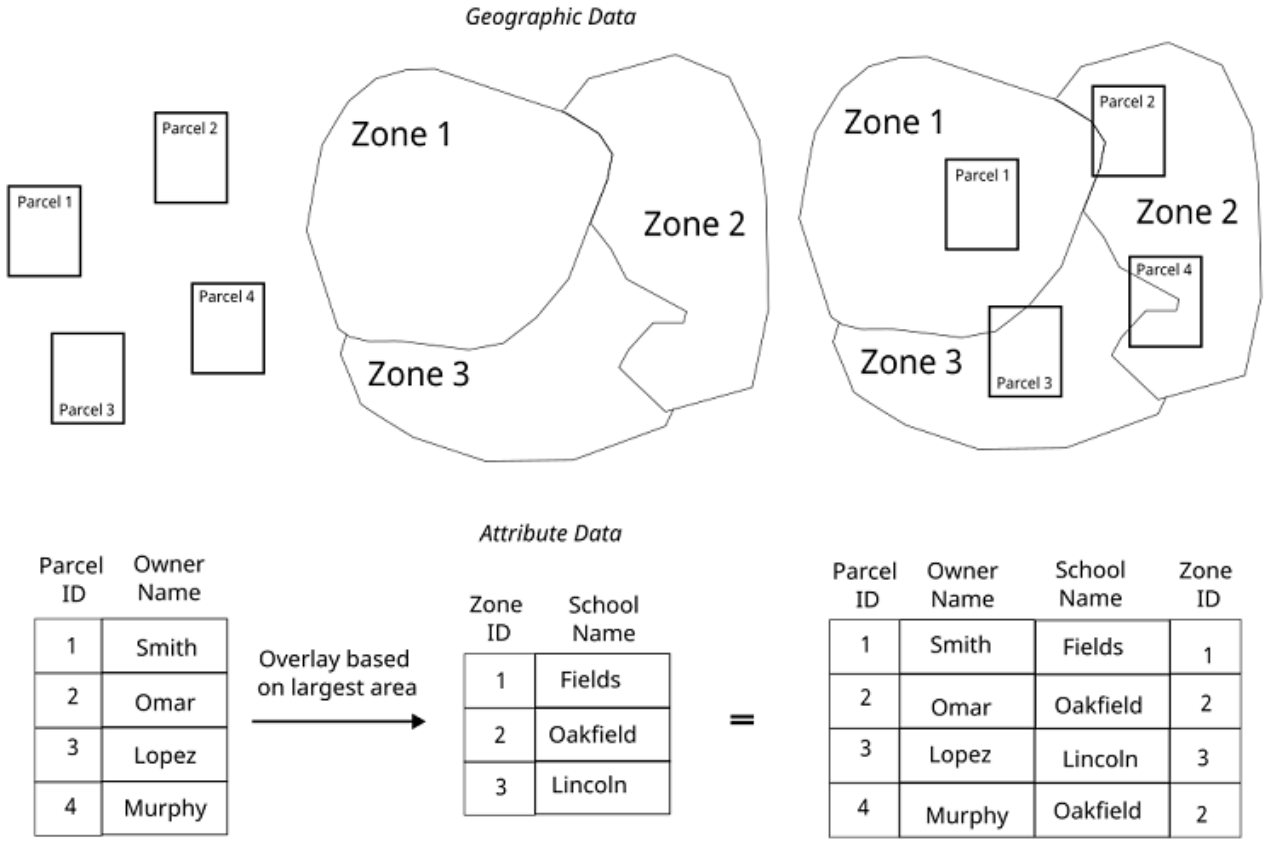

Figure 4. Example concept for polygon-to-polygon spatial join. Source: author.

However, suppose the spatial predicate specified that the logical condition of a match is based on the centroid of the parcel being contained in a given zone polygon. In that case, the results will vary, such as Parcel ID 4 falling within the bound of Zone ID 2.

4.2.2 Point to Polygon

Another spatial join operation that addresses the geographic question of contains considers what points are contained within the bound of which polygons. Because lines (1D) are typically stored in a GIS as a series of connected points, similar spatial join operations are available for geographic features represented in this way (e.g., a road or a powerline). Topological comparisons such as inside intersect and adjacent can compare lines and polygons in a spatial join operation.

Consider the example of mapping the police calls for service within a set of policing districts or zones. The calls for service are recorded as points, and the policing zones are polygons. Before carrying out any spatial join operation, it is helpful to consider the relationship

between these two datasets. The diagram in Fig. 5 shows that one police zone may contain zero-to-many calls for service.

Consider the case of police calls for services (represented as points) and police zones (represented as polygons). The spatial relationship between these entities is illustrated in Fig. 5.

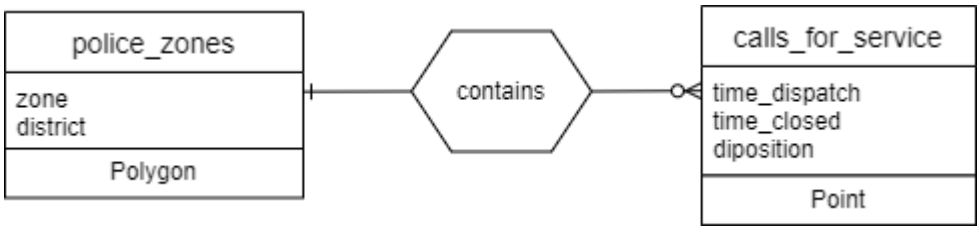

Figure 5. Example ERD contains a relationship. Source: author.

In this ERD, the (spatial) cardinality dictates that police zone boundaries contain calls for service spatially. Also, important to note is a one-to-many relationship cardinality between police zones and calls for service. Because of this, it will make sense to see the spatial relationship between police zones and calls for service contains. The relational algebra expression for a spatial join that is used to query all of the calls for service (calls_for_service) that is contained in each specific zone might look like this:

contains(police_zones, calls_for_service)

This spatial join could be used to answer geographic questions as determining which district has the most calls for service during specific periods.

4.3.4 Raster

Raster data is stored as a contiguous grid (called cells) with a specified spatial resolution and extent. Geographic features best represented by gradual changes over space, such as vegetation, elevation, or temperature, are typically stored in this data format. A spatial join between two distinct raster datasets is most efficiently handled between datasets that match in extent and cell resolution. When raster data that does not match in extent or resolution needs to be joined, then topological properties may be utilized to perform additional spatial operations to bring them into a similar extent/resolution (e.g., resampling).

Conceptually the joining operations for raster are similar to those carried out on vector data. However, the GIS implementation will vary due to the unique aspect of how raster data is stored and processed. For instance, raster data typically use the smallest required data size (e.g., unsigned bytes). There may be a many-to-one relationship between raster cells and attribute rows, as discussed in Bolstad (2019).

Consider the need to join elevation data with land cover data (Fig. 6). In this case, we can perform a raster overlay, which performs a cell-by-cell combination as a spatial join. Let's consider the spatial join (via overlay) of two nominal geographic data, such as soil class and land cover. We can see a combination of cell values in the resulting raster attribute tables.

However, when joining raster data that contains ratio, interval, or continuous data, the results of a raster spatial join are usually handled using map algebra within GIS. For an expanded discussion of raster operations and map algebra, see Cai (2022).

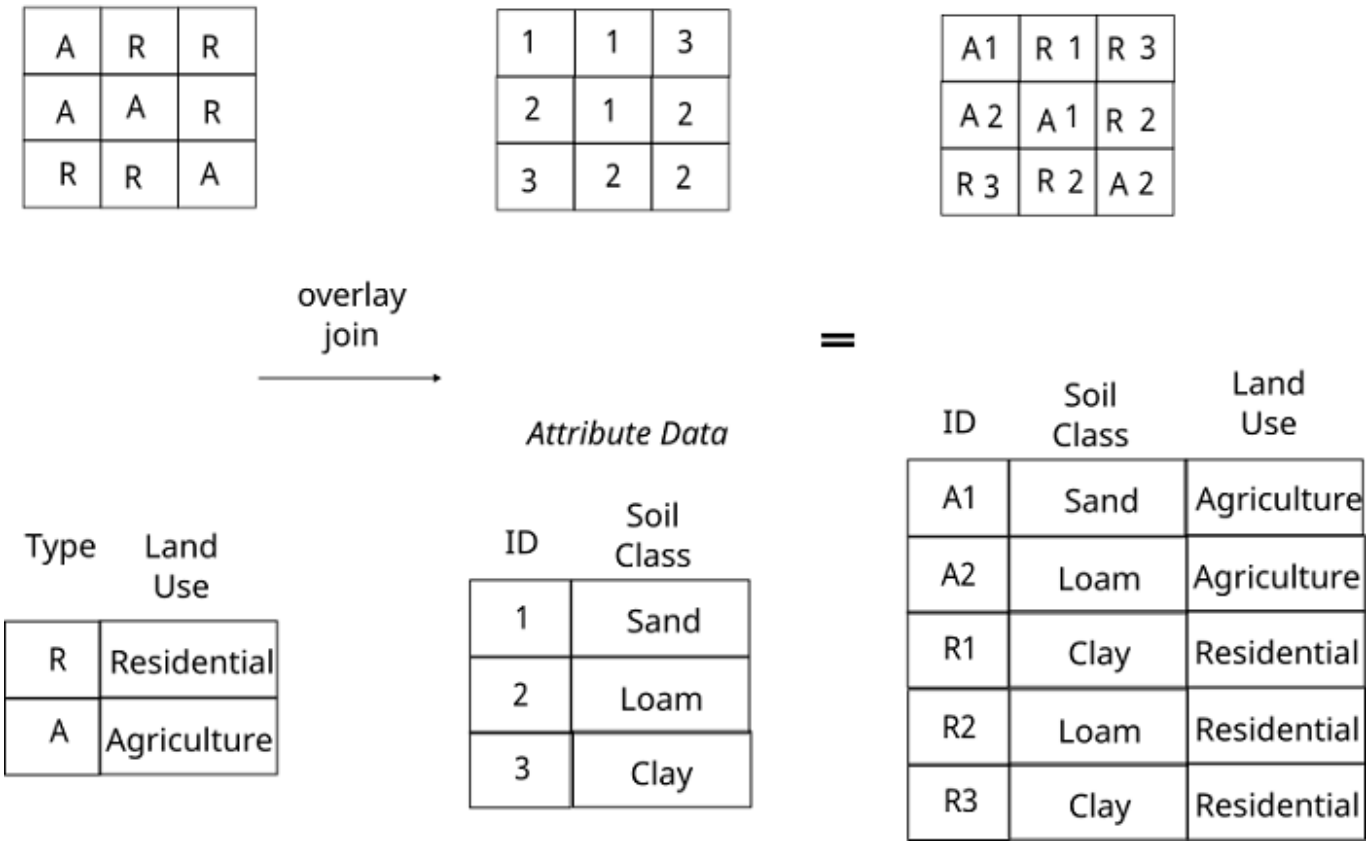

Figure 6. Example raster spatial join. Source: author.

5. Challenges with Spatial Joins

Because a spatial join can be computationally intensive, methods for developing efficient methods to process spatial intersection joins for two-dimensional data sets efficiently have evolved (Belussi et al., 2004; Faloutsos et al., 2000). For instance, a minimum bounding rectangle (MBR) is the smallest rectangular box that encloses the object (Hsu & Obe, 2021) and can simplify or optimize spatial join and associated queries. MBR coordinates for features can be stored as an index to limit the computational cost of spatial joins. For instance, an MBR index can limit the number of features on a first pass and then do a more detailed search on testing for spatial intersections based on individual vertices. This MBR method then uses a multi-step process where on a filter step a query is applied on objects representing a bounding rectangle (Mamoulis, 2011). The MBR is then used in subsequent steps to prune or refine the search space, which can speed up computationally-intensive operations such as a spatial join.

A spatial join can only handle many-to-one cardinality. This limitation is essential to geographic problem solving (e.g., allowing spatial integration of data of different geometries and area units to pre-determined ones for spatial analysis). Bolstad (2019) describes this limitation by explaining that the results of a many-to-many join should be

avoided due to ambiguity in analyzing the results table. Further, the conventional spatial join can only handle geographic features of simplexes (i.e., a feature is a point, a line, or a polygon) but can be problematic on complexes (i.e., multi-point features, multi-line features, or multi-polygon features) due to the limitation of this one-to-many cardinality.

References

Beeri, C., Kanza, Y., Safra, E., & Sagiv, Y. (2004). Object fusion in geographic information systems. In Proceedings of the Thirtieth International Conference on Very Large Data Bases. 30:816-827.

Belussi, A., Bertino, E., & Nucita, A. (2004). Grid based methods for estimating spatial join selectivity. In Proceedings of the 12th annual ACM International Workshop on Geographic Information Systems (pp. 92-100).

Bolstad, P. (2019). GIS Fundamentals: A First Text on Geographic Information Systems, 6th Edition. Acton, MA: XanEdu Publishing Inc.

Cai, H. (2022). Overlay. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2022 Edition), John P. Wilson (ed.).

Calkins, H. W. (1996). Entity relationship modeling of spatial data for geographic information systems. International Journal of Geographical Information Systems, 10(1).

Chan, E.P.F. and Ng, J.N.H. (1997). A general and efficient implementation of geometric operators and predicates. In: Scholl, M., Voisard, A. (eds) Advances in Spatial Databases. SSD 1997. Lecture Notes in Computer Science, vol 1262. Springer, Berlin, Heidelberg.

Cliff, A. D., & Ord, J. K. (1975). Model Building and the Analysis of Spatial Pattern in Human Geography. Journal of the Royal Statistical Society: Series B (Methodological), 37(3), 297-328.

Codd, E. F. (1980). Data models in database management. ACM SIGMOD Record - Proceedings of the workshop on Data abstraction, databases and conceptual modelling, 11(2), 112-114.

Diamond, L. (2019). Vector Formats and Sources. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2019 Edition), John P. Wilson (ed.).

Egenhofer, M. J., Glasgow, J., Gunther, O., Herring, J. R., & Peuquet, D. J. (1999). Progress in computational methods for representing geographical concepts. International Journal of Geographical Information Science, 13(8), 775-796.

Faloutsos, C., Seeger, B., Traina, A., & Traina Jr, C. (2000). Spatial join selectivity using power laws. In Proceedings of the 2000 ACM SIGMOD international conference on Management of data (pp. 177-188).

Hachadoorian, L. (2019). SQL Languages for GIS. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2019 Edition), John P. Wilson (Ed).

Mamoulis, N. (2011). Spatial data management. Synthesis Lectures on Data Management, 3(6), 1-149.

Nyerges, T. L. (2017). Conceptual Data Models. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2017 Edition), John P. Wilson (ed.).

Nyerges, T. L. (2017a). Logical Data Models. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2017 Edition), John P. Wilson (ed.).

Nyerges, T. L. (2017b). Physical Data Models. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2017 Edition), John P. Wilson (ed.).

Obe, R. and Hsu, L. S. (2021). PostGIS in Action. Simon and Schuster.

Williams, C. (2019). Raster Formats and Sources. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2019 Edition), John P. Wilson (Ed.).

Worboys, M. F., & Duckham, M. (2004). GIS: a Computing Perspective (2nd Edition). CRC Press.